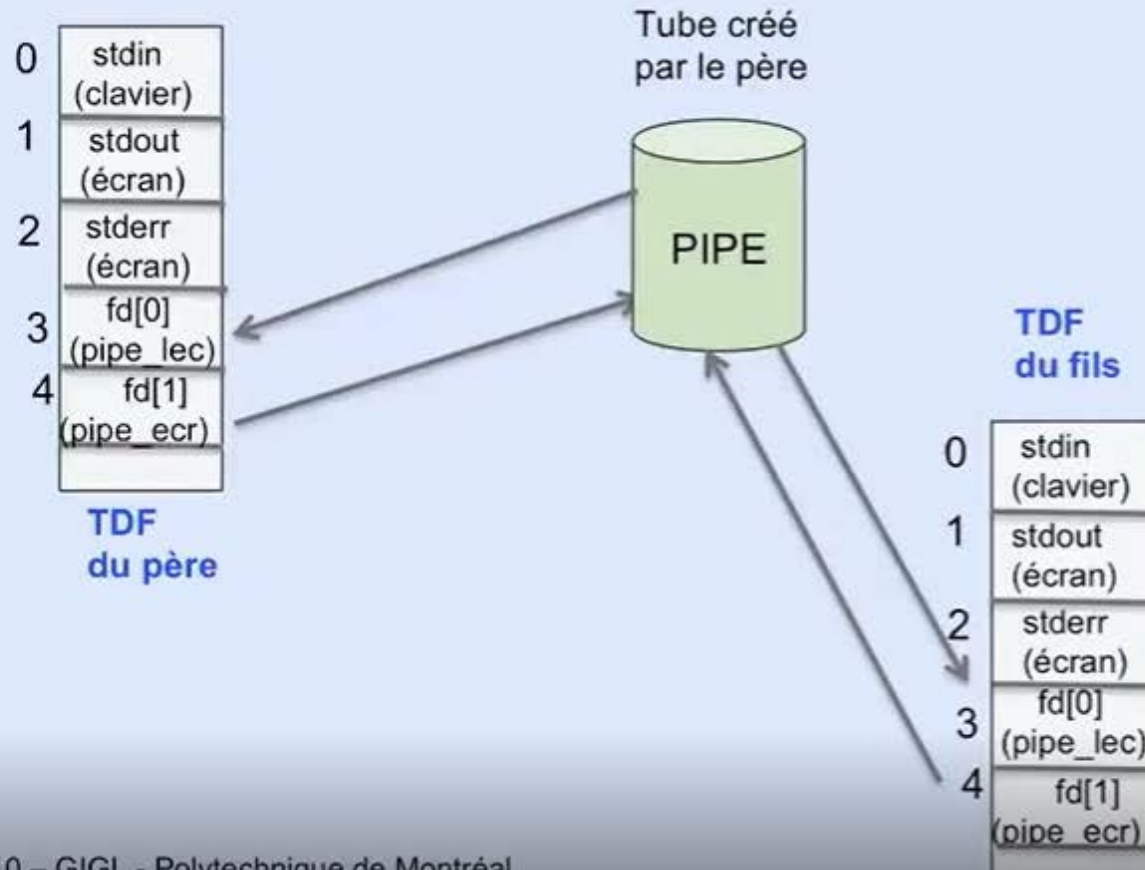


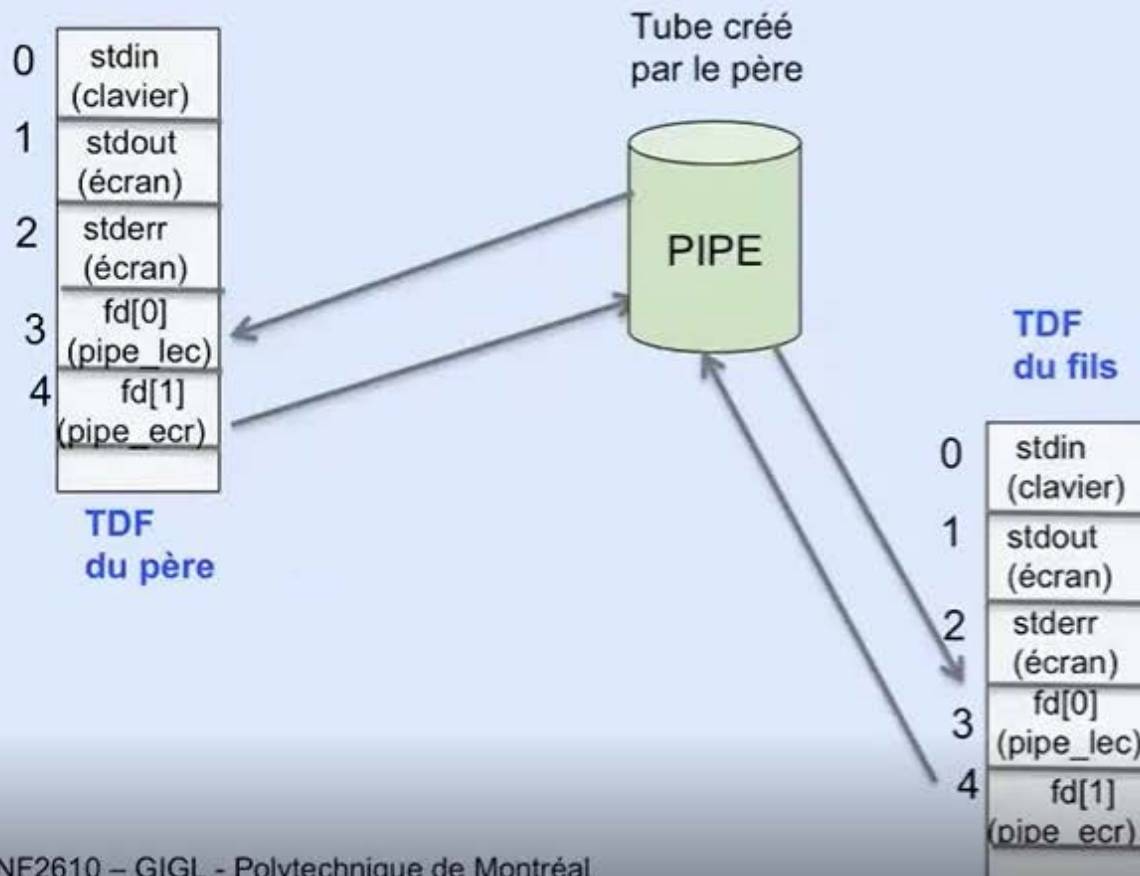
# Exemple 1 – Communication père-fils via un tube anonyme



```
int main() { int fd[2]; pipe(fd);
  if (fork() == 0) {
    close(fd[1]); char buf;
    while (read(fd[0], &buf, 1) > 0) { write(1, &buf, 1); }
    close(fd[0]);
  } else {
    close(fd[0]);
    write(fd[1], "Je te salue mon fils.\n", 22);
    close(fd[1]);
    wait(NULL); }
  _exit(0);
}
```

Remarque :  
Après la création du fils, les processus père et fils évoluent en concurrence → différents entrelacements de leurs instructions sont possibles.

# Exemple 1 – Communication père-fils via un tube anonyme



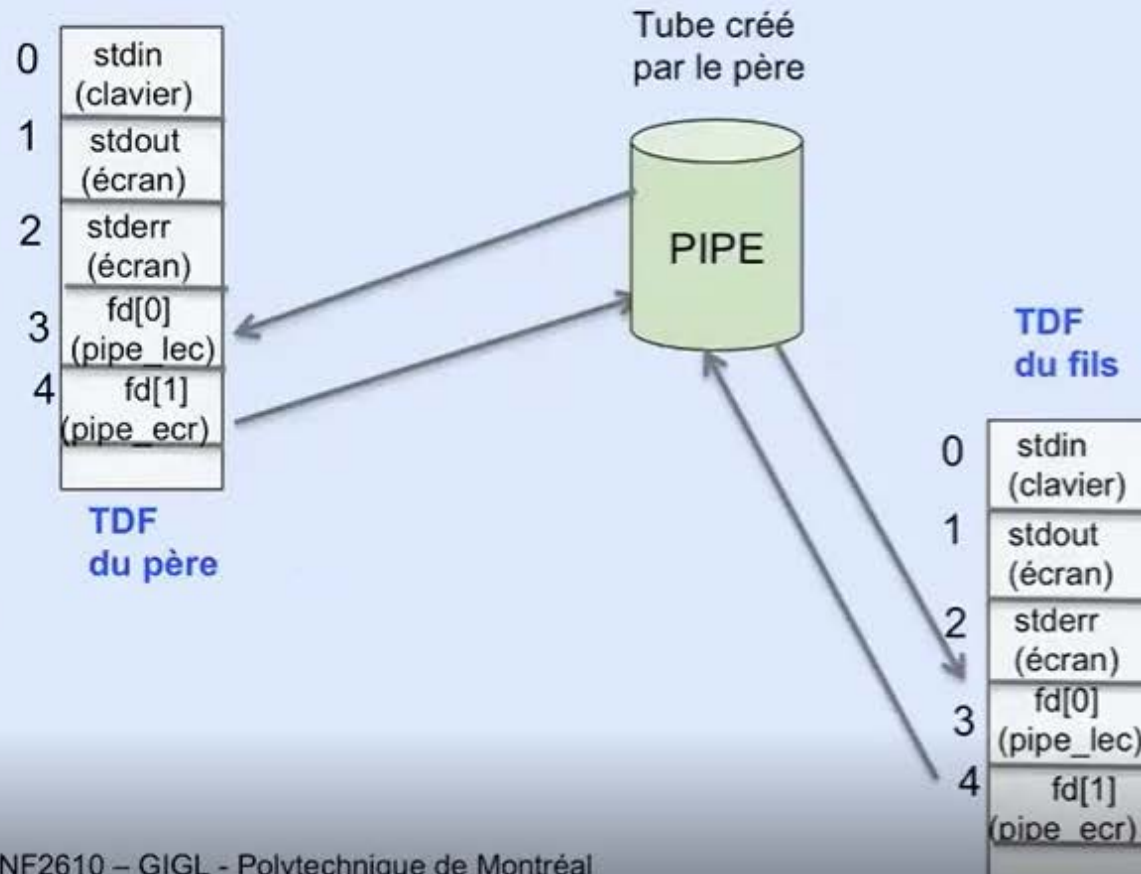
```
int main() { int fd[2]; pipe(fd);
  if (fork() == 0) {
    close(fd[1]); char buf;
    while (read(fd[0], &buf, 1) > 0) { write(1, &buf, 1); }
    close(fd[0]);
  } else {
    close(fd[0]);
    write(fd[1], "Je te salue mon fils.\n", 22);
    close(fd[1]);
    wait(NULL); }
  _exit(0);
}
```

Un entrelacement possible :

```
F: close(fd[1])
F: read(fd[0], &buf, 1)
P: close(fd[0])
P: write(fd[1], ".....", 22);
P: close(fd[1]);
P: wait (NULL);
F: while (read(fd[0], &buf, 1) > 0)
  { write(1, &buf, 1); }
F: close(fd[0]);
F: _exit(0);
P: poursuit wait(NULL);
P: _exit(0);
```

# Exemple 1 – Communication père-fils via un tube anonyme

```
int main() { int fd[2]; pipe(fd);
  if (fork() == 0) {
    close(fd[1]); char buf;
    while (read(fd[0], &buf, 1) > 0) { write(1, &buf, 1); }
    close(fd[0]);
  } else {
    close(fd[0]);
    write(fd[1], "Je te salue mon fils.\n",22);
    close(fd[1]);
    wait(NULL); }
  _exit(0);
}
```

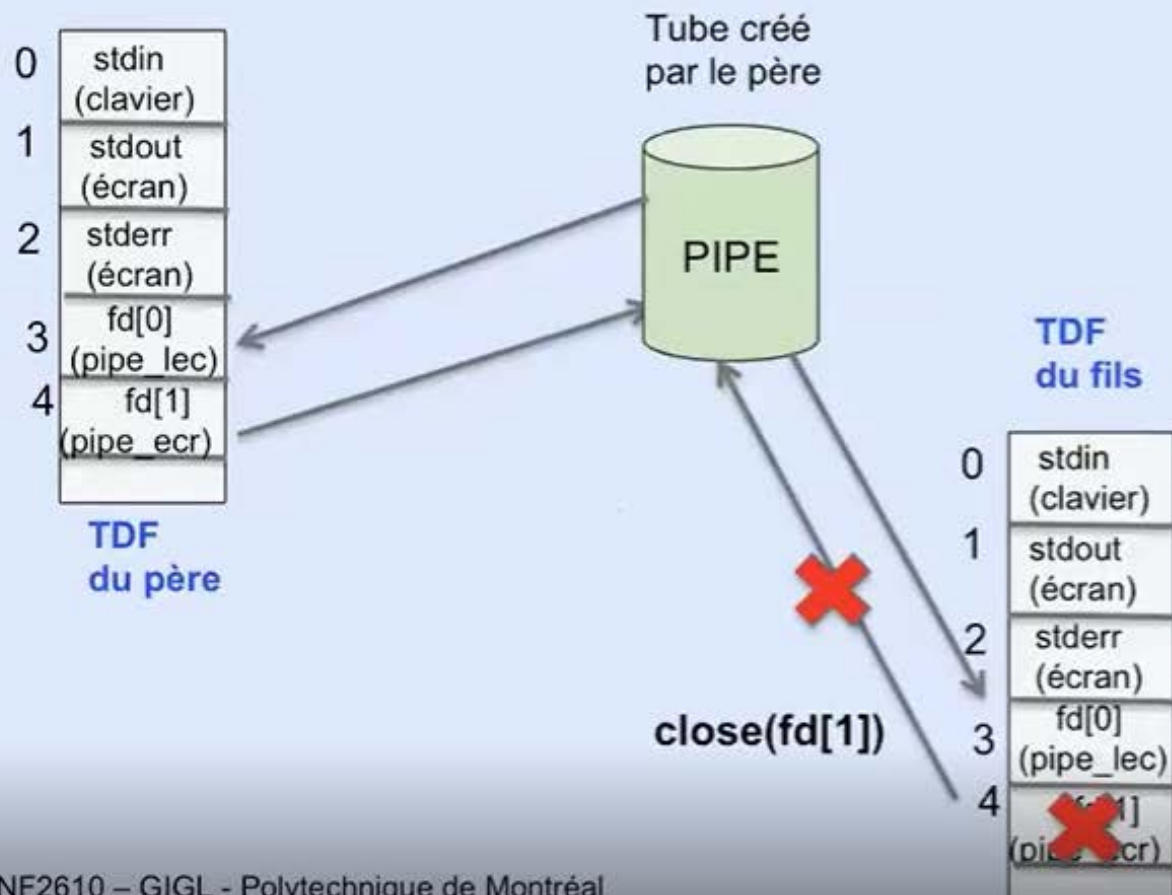


Un entrelacement possible :

```
F: close(fd[1])
F: read(fd[0], &buf, 1)
P: close(fd[0])
P: write(fd[1], ".....",22);
P: close(fd[1]);
F: while (read(fd[0],&buf,1)>0)
  { write(1, &buf, 1); }
F: close(fd[0]);
F: _exit(0);
P: poursuit wait(NULL);
P: _exit(0);
```



# Exemple 1 – Communication père-fils via un tube anonyme

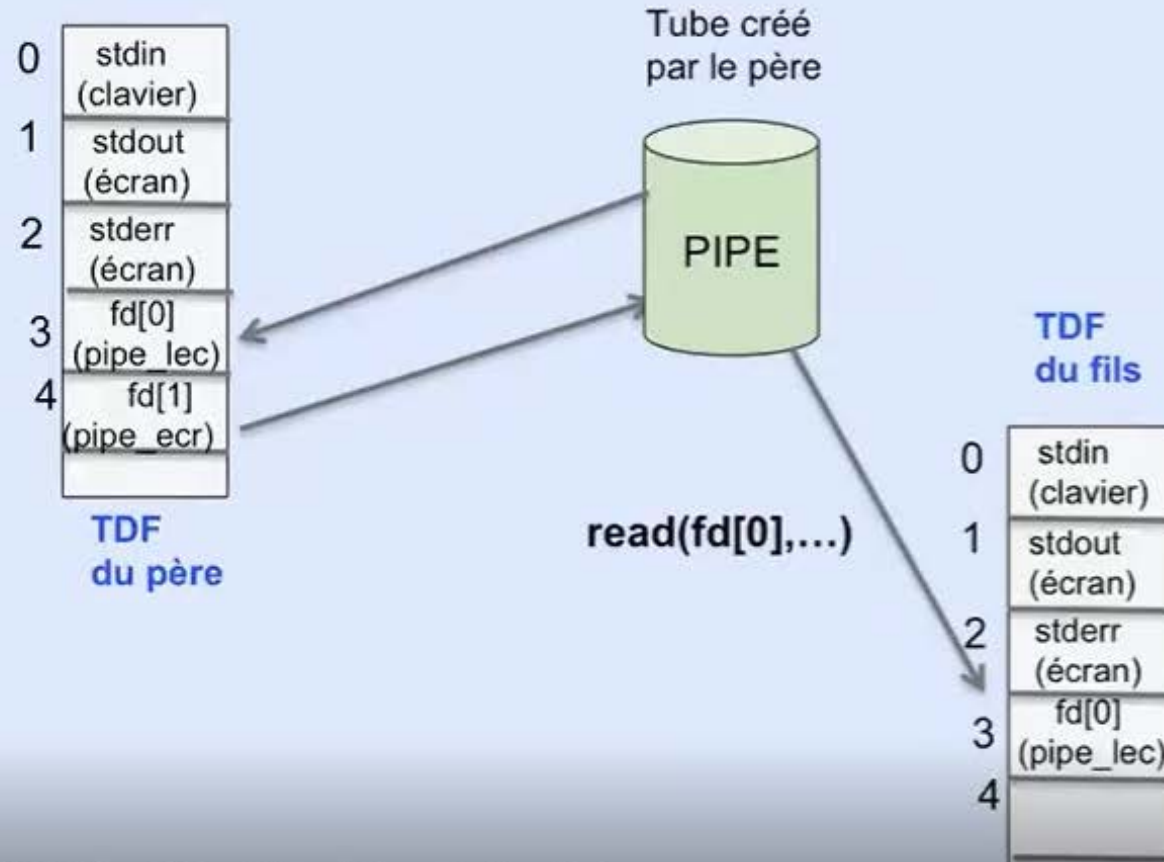


```
int main() { int fd[2]; pipe(fd);
  if (fork() == 0) {
    close(fd[1]); char buf;
    while (read(fd[0], &buf, 1) > 0) { write(1, &buf, 1); }
    close(fd[0]);
  } else {
    close(fd[0]);
    write(fd[1], "Je te salue mon fils.\n", 22);
    close(fd[1]);
    wait(NULL); }
  _exit(0);
}
```

Un entrelacement possible :

```
F: close(fd[1])
F: read(fd[0], &buf, 1)
P: close(fd[0])
P: write(fd[1], ".....", 22);
P: close(fd[1]);
P: wait (NULL);
F: while (read(fd[0], &buf, 1) > 0)
  { write(1, &buf, 1); }
F: close(fd[0]);
F: _exit(0);
P: poursuit wait(NULL);
P: _exit(0);
```

# Exemple 1 – Communication père-fils via un tube anonyme



```
int main() { int fd[2]; pipe(fd);
  if (fork() == 0) {
    close(fd[1]); char buf;
    while (read(fd[0], &buf, 1) > 0) { write(1, &buf, 1); }
    close(fd[0]);
  } else {
    close(fd[0]);
    write(fd[1], "Je te salue mon fils.\n", 22);
    close(fd[1]);
    wait(NULL); }
  _exit(0);
}
```

Un entrelacement possible :

F: **close(fd[1])**

F: **read(fd[0], &buf, 1)** → l'état bloqué

P: close(fd[0])

P: write(fd[1], ".....", 22);

P: close(fd[1]);

P: wait (NULL);

F: while (read(fd[0], &buf, 1) > 0) { write(1, &buf, 1); }

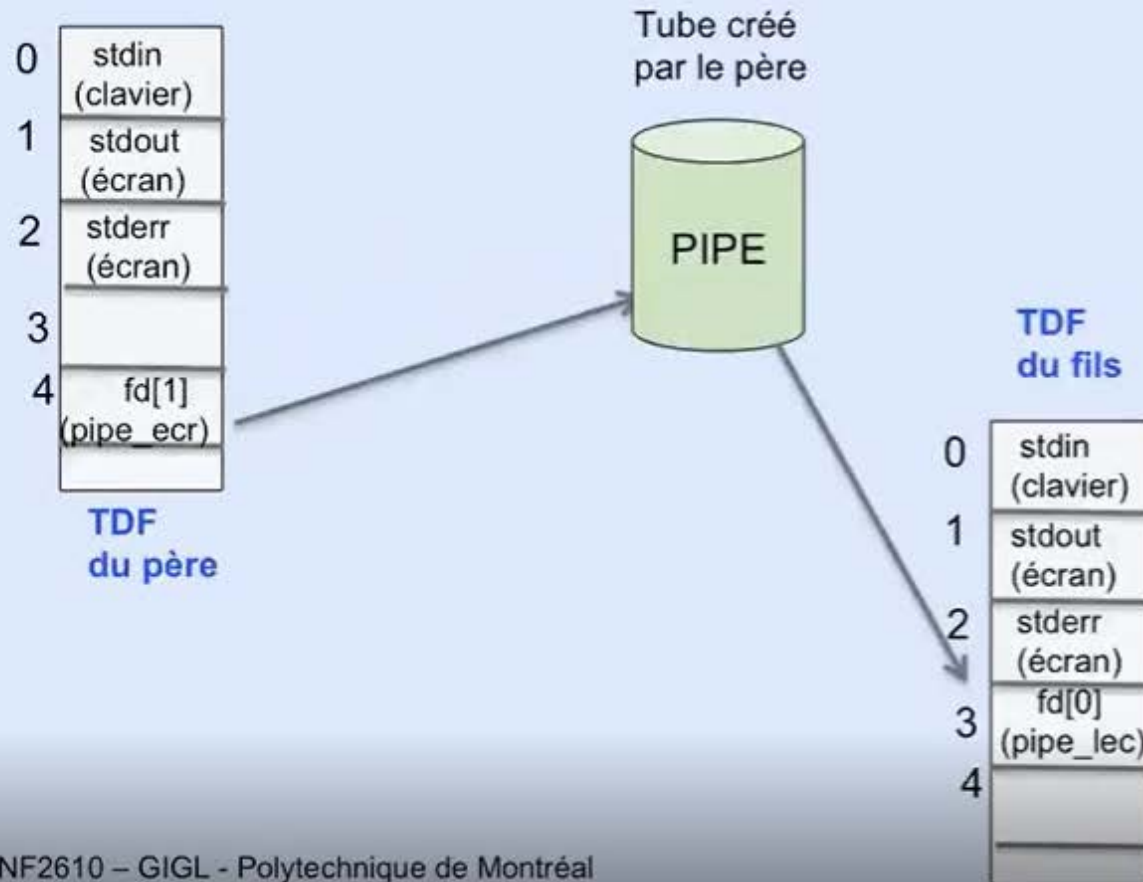
F: close(fd[0]);

F: \_exit(0);

P: poursuit wait(NULL);

P: \_exit(0);

# Exemple 1 – Communication père-fils via un tube anonyme



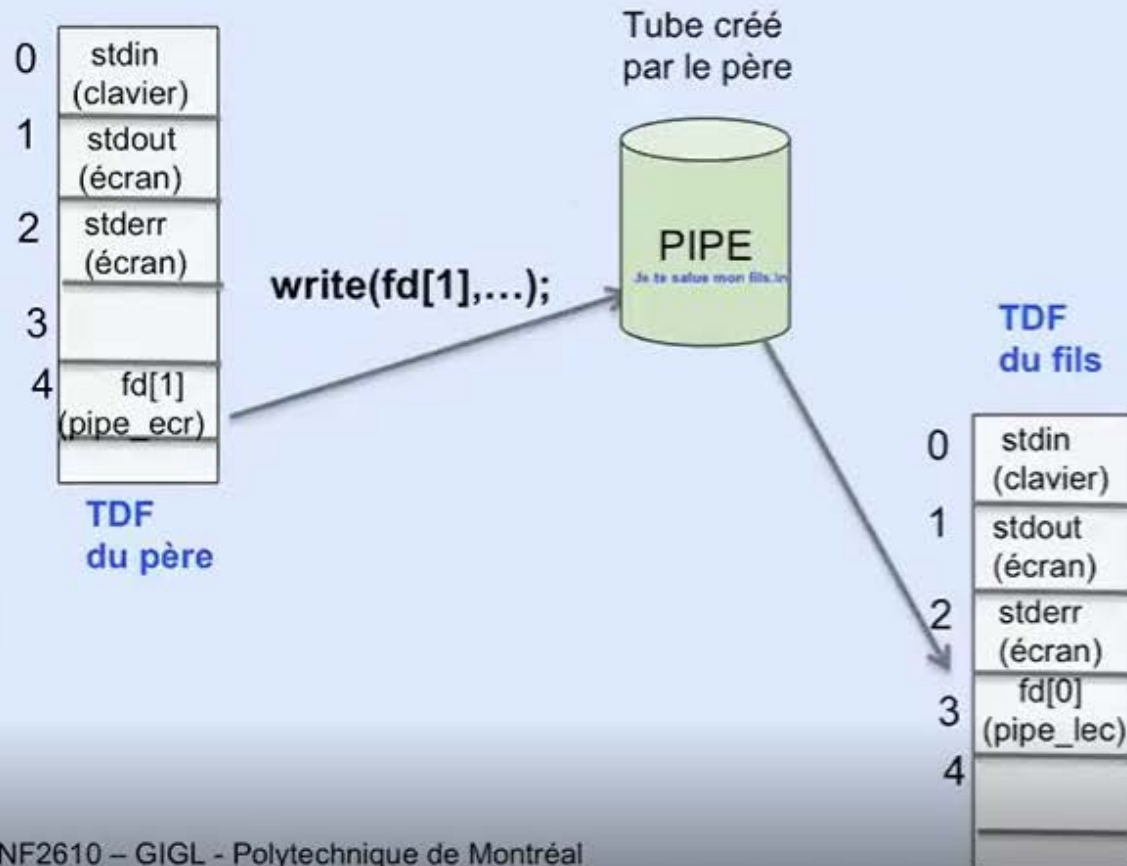
```
int main() { int fd[2]; pipe(fd);
  if (fork() == 0) {
    close(fd[1]); char buf;
    while (read(fd[0], &buf, 1) > 0) { write(1, &buf, 1); }
    close(fd[0]);
  } else {
    close(fd[0]);
    write(fd[1], "Je te salue mon fils.\n", 22);
    close(fd[1]);
    wait(NULL); }
  _exit(0);
}
```

Un entrelacement possible :

F: **close(fd[1])**  
 F: **read(fd[0], &buf, 1) → l'état bloqué.**  
 P: **close(fd[0])**  
 P: **write(fd[1], ".....", 22)**  
 P: close(fd[1])  
 P: wait (NULL)  
 F: while (read(fd[0], &buf, 1) > 0)  
     { write(1, &buf, 1); }  
 F: close(fd[0]);  
 F: \_exit(0);  
 P: poursuit wait(NULL);  
 P: \_exit(0);



# Exemple 1 – Communication père-fils via un tube anonyme



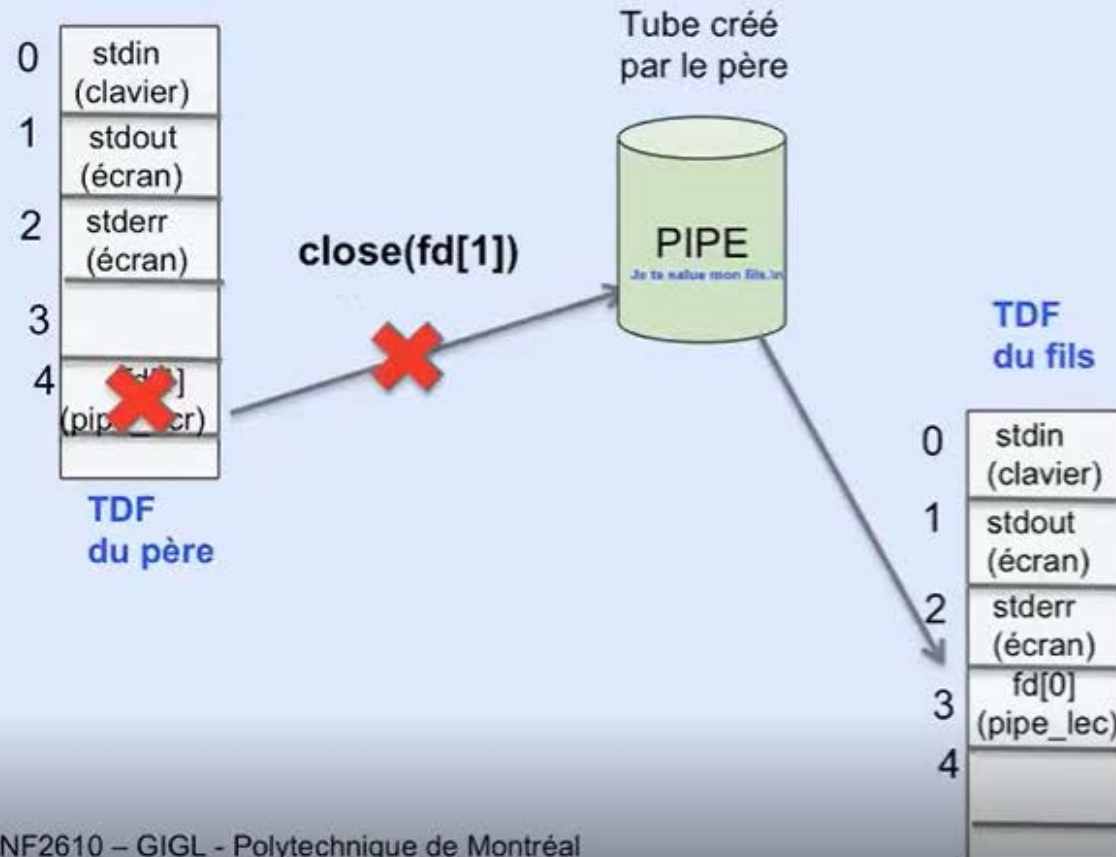
```
int main() { int fd[2]; pipe(fd);
  if (fork() == 0) {
    close(fd[1]); char buf;
    while (read(fd[0], &buf, 1) > 0) { write(1, &buf, 1); }
    close(fd[0]);
  } else {
    close(fd[0]);
    write(fd[1], "Je te salue mon fils.\n",22);
    close(fd[1]);
    wait(NULL); }
  _exit(0);
}
```

Un entrelacement possible :

F: `close(fd[1])`  
 F: `read(fd[0], &buf, 1)` → l'état bloqué.  
 P: `close(fd[0])`  
 P: `write(fd[1], ".....",22)` → débloque F  
 P: `close(fd[1])`  
 P: `wait (NULL)`  
 F: `while (read(fd[0],&buf,1)>0)`  
     `{ write(1, &buf, 1); }`  
 F: `close(fd[0]);`  
 F: `_exit(0);`  
 P: poursuit `wait(NULL);`  
 P: `_exit(0);`

# Exemple 1 – Communication père-fils via un tube anonyme

```
int main() { int fd[2]; pipe(fd);
  if (fork() == 0) {
    close(fd[1]); char buf;
    while (read(fd[0], &buf, 1) > 0) { write(1, &buf, 1); }
    close(fd[0]);
  } else {
    close(fd[0]);
    write(fd[1], "Je te salue mon fils.\n",22);
    close(fd[1]);
    wait(NULL); }
  _exit(0);
}
```



Un entrelacement possible :

F: **close(fd[1])**  
 F: **read(fd[0], &buf, 1) → l'état bloqué.**  
 P: **close(fd[0])**  
 P: **write(fd[1], ".....",22) → débloque F**  
 P: **close(fd[1])**  
 P: wait (NULL)  
 F: while (read(fd[0],&buf,1)>0)  
     { write(1, &buf, 1); }  
 F: close(fd[0]);  
 F: \_exit(0);  
 P: poursuit wait(NULL);  
 P: \_exit(0);



# Exemple 1 – Communication père-fils via un tube anonyme

0	stdin (clavier)
1	stdout (écran)
2	stderr (écran)
3	
4	

TDF  
du père

Tube créé  
par le père



TDF  
du fils

0	stdin (clavier)
1	stdout (écran)
2	stderr (écran)
3	fd[0] (pipe lec)
4	

```
int main() { int fd[2]; pipe(fd);
  if (fork() == 0) {
    close(fd[1]); char buf;
    while (read(fd[0], &buf, 1) > 0) { write(1, &buf, 1); }
    close(fd[0]);
  } else {
    close(fd[0]);
    write(fd[1], "Je te salue mon fils.\n", 22);
    close(fd[1]);
    wait(NULL); }
  _exit(0);
}
```

Un entrelacement possible :

F: **close(fd[1])**

F: **read(fd[0], &buf, 1) → l'état bloqué.**

P: **close(fd[0])**

P: **write(fd[1], ".....", 22) → débloque F**

P: **close(fd[1])**

P: **wait (NULL) → l'état bloqué**

F: while (read(fd[0], &buf, 1) > 0)  
{ write(1, &buf, 1); }

F: close(fd[0]);

F: **\_exit(0);**

P: poursuit wait(NULL);

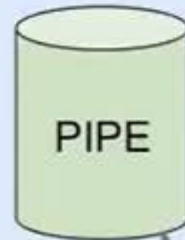
P: **\_exit(0);**

# Exemple 1 – Communication père-fils via un tube anonyme

0	stdin (clavier)
1	stdout (écran)
2	stderr (écran)
3	
4	

TDF  
du père

Tube créé  
par le père



fin de fichier

TDF  
du fils

0	stdin (clavier)
1	stdout (écran)
2	stderr (écran)
3	fd[0] (pipe lec)
4	

```
int main() { int fd[2]; pipe(fd);
  if (fork() == 0) {
    close(fd[1]); char buf;
    while (read(fd[0], &buf, 1) > 0) { write(1, &buf, 1); }
    close(fd[0]);
  } else {
    close(fd[0]);
    write(fd[1], "Je te salue mon fils.\n",22);
    close(fd[1]);
    wait(NULL); }
  _exit(0);
}
```

Un entrelacement possible :

F: **close(fd[1])**

F: **read(fd[0], &buf, 1) → l'état bloqué.**

P: **close(fd[0])**

P: **write(fd[1], ".....",22) → débloque F**

P: **close(fd[1])**

P: **wait (NULL) → l'état bloqué**

F: **while (read(fd[0],&buf,1)>0) { write(1, &buf, 1); }** 2 conditions pour sortir du while:  
1) tous les écrivains ont fait des close et  
2) le pipe est vide

Autrement read retourne alors 0

F: **close(fd[0]);**

F: **\_exit(0);**

P: **poursuit wait(NULL);**

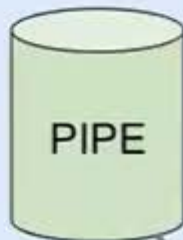
P: **\_exit(0);**

# Exemple 1 – Communication père-fils via un tube anonyme

0	stdin (clavier)
1	stdout (écran)
2	stderr (écran)
3	
4	

TDF du père

Tube créé par le père



close(fd[0]);



TDF du fils

0	stdin (clavier)
1	stdout (écran)
2	stderr (écran)
3	fd[0] (pipe lec)
4	

```
int main() { int fd[2]; pipe(fd);
  if (fork() == 0) {
    close(fd[1]); char buf;
    while (read(fd[0], &buf, 1) > 0) { write(1, &buf, 1); }
    close(fd[0]);
  } else {
    close(fd[0]);
    write(fd[1], "Je te salue mon fils.\n", 22);
    close(fd[1]);
    wait(NULL); }
  _exit(0);
}
```

Un entrelacement possible :

F: **close(fd[1])**  
 F: **read(fd[0], &buf, 1)** → l'état bloqué.  
 P: **close(fd[0])**  
 P: **write(fd[1], ".....", 22)** → débloque F  
 P: **close(fd[1])**  
 P: **wait (NULL)** → l'état bloqué  
 F: **while (read(fd[0], &buf, 1) > 0)**  
     **{ write(1, &buf, 1); }**  
 F: **close(fd[0]);**  
 F: **\_exit(0);**  
 P: poursuit wait(NULL);  
 P: **\_exit(0);**

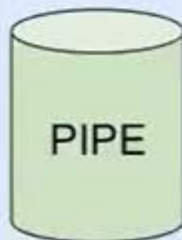


# Exemple 1 – Communication père-fils via un tube anonyme

0	stdin (clavier)
1	stdout (écran)
2	stderr (écran)
3	
4	

TDF  
du père

Tube créé  
par le père



TDF  
du fils

0	stdin (clavier)
1	stdout (écran)
2	stderr (écran)
3	
4	

```
int main() { int fd[2]; pipe(fd);
  if (fork() == 0) {
    close(fd[1]); char buf;
    while (read(fd[0], &buf, 1) > 0) { write(1, &buf, 1); }
    close(fd[0]);
  } else {
    close(fd[0]);
    write(fd[1], "Je te salue mon fils.\n", 22);
    close(fd[1]);
    wait(NULL); }
  _exit(0);
}
```

Un entrelacement possible :

F: **close(fd[1])**

F: **read(fd[0], &buf, 1) → l'état bloqué.**

P: **close(fd[0])**

P: **write(fd[1], ".....", 22) → débloquent F**

P: **close(fd[1])**

P: **wait (NULL) → l'état bloqué**

F: **while (read(fd[0], &buf, 1) > 0)**  
**{ write(1, &buf, 1); }**

F: **close(fd[0]);**

F: **\_exit(0); → débloquent P**  
**→ F passe à l'état zombie**

P: poursuit wait(NULL);

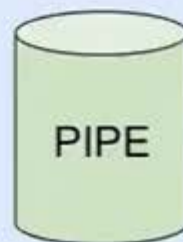
P: **\_exit(0);**

# Exemple 1 – Communication père-fils via un tube anonyme

0	stdin (clavier)
1	stdout (écran)
2	stderr (écran)
3	
4	

TDF  
du père

Tube créé  
par le père



TDF  
du fils

0	stdin (clavier)
1	stdout (écran)
2	stderr (écran)
3	
4	

```
int main() { int fd[2]; pipe(fd);
  if (fork() == 0) {
    close(fd[1]); char buf;
    while (read(fd[0], &buf, 1) > 0) { write(1, &buf, 1); }
    close(fd[0]);
  } else {
    close(fd[0]);
    write(fd[1], "Je te salue mon fils.\n", 22);
    close(fd[1]);
    wait(NULL); }
  _exit(0);
}
```

Un entrelacement possible :

F: **close(fd[1])**

F: **read(fd[0], &buf, 1) → l'état bloqué.**

P: **close(fd[0])**

P: **write(fd[1], ".....", 22) → débloquent F**

P: **close(fd[1])**

P: **wait (NULL) → l'état bloqué**

F: **while (read(fd[0], &buf, 1) > 0)**  
**{ write(1, &buf, 1); }**

F: **close(fd[0]);**

F: **\_exit(0); → débloquent P**

**→ F passe à l'état zombie**

P: **poursuit wait(NULL); → F quitte le système**

P: **\_exit(0);**



## Exemple 1' – Attention aux interblocages!

```
int main() { int fd[2]; pipe(fd);  
    if (fork() == 0) {  
        close(fd[1]); char buf;  
        while (read(fd[0], &buf, 1) > 0)  
            { write(1, &buf, 1); }  
        close(fd[0]);  
    } else {  
        close(fd[0]);  
        write(fd[1], "Je te salue mon fils.\n", 22);  
        // close(fd[1]);  
        wait(NULL); }  
    _exit(0);  
}
```

Attention! Si l'écrivain ne ferme pas, les conditions du while dans le fils ne seront jamais rencontrés, car le Père est bloqué sur wait(NULL) à attendre la fin du fils.